A SOFTWARE FRAMEWORK FOR BLAST EVENT SIMULATION

D.A. Swensen, M.K. Denison
Reaction Engineering International
Salt Lake City, UT 84101

James Guilkey, Todd Harman
University of Utah
Salt Lake City, UT 84112

Richard Goetz
US Army RDECOM-TARDEC
Warren, MI 48397

ABSTRACT

Protection of US Army vehicles and personnel against landmine and IED threats is an increasingly important concern in the area of defense research. In this paper we describe the development of a Blast Computational Framework (BCF) that will provide an advanced modeling environment and a suite of tools for performing soil bound explosion simulations and their effects on vehicles and on the human occupants of the vehicles. The BCF will provide a virtual test-bed where disparate computational models can seamlessly interact with one another to provide a unified modeling solution for blast-vehicle-occupant scenarios. The BCF is being developed using state-of-the-art component-based software architecture and will provide a suite of integrated models consisting of both new and existing simulation tools. The enhanced simulation capabilities provided by the BCF will serve to better protect the crews of existing vehicles and to help design next-generation vehicles as well.

1. INTRODUCTION

To adapt to the modern battlefield, military forces are coming to rely more heavily on the mobility and rapid deployment of lightly armored vehicles [Global Security]. These vehicles, however, are faced with a growing threat from highly lethal anti-tank, anti-personnel, and an increased number of improvised explosive devices such as those so predominantly used to disrupt peace-keeping efforts in Iraq. Better understanding of landmine detonation forces, loading conditions and vehicle response is vital in designing more resilient armored vehicles to minimize human and equipment losses in next-generation military forces.

Through funding from a DoD SBIR award, Reaction Engineering International (REI) is developing advanced software tools to facilitate blast-vehicle-crew simulation studies. The approach taken in the project consists of two major efforts. First, a computational framework is being developed that will allow integration of the various disparate computational models required for a complete blast-to-crew simulation. Second, a state-of-the-art blast-structure model is being enhanced with soil models so that it can be used to simulate the blast and soil portion of the event.

The BCF computational framework is being developed using component-based software techniques to provide for a high-level of interoperability, programming language independence and computer network transparency. The BCF will serve as a computational canvas where component models required for a simulation can be graphically instantiated, interconnected and executed. It will also provide embedded scientific visualization capabilities for simulation results.

The blast and soil modeling is being performed using MPMICE, an advanced new simulation tool being developed as part of the DoE ASCI program at the University of Utah (C-SAFE). The modeling techniques used by MPMICE differ from traditional methods, and hold promise for increased accuracy. The vehicles in the simulations will be modeled using the MPMICE MPM techniques (for approximate vehicle models), DYNA3D FE code and off-the-shelf packages such as LS-DYNA.

The project work scope includes integrating a hierarchy of blast simulation tools, ranging from reduced-order models and decoupled preprocessor-based models, to a comprehensive, tightly coupled simulation tool based on the C-SAFE and DYNA3D codes. The model hierarchy will allow for blast simulations to be performed with appropriate levels of detail, depending on the required accuracy, ease of use, simulation run-time and available computational resources.

In the following we provide, in order, our approach for modeling the blast-vehicle scenario, a description of the computational framework software under development and an overview of how the various tools have been applied to date.

2. MODELING APPROACH

While the BCF framework has the extensibility to integrate nearly any type of computational model, the scope of the SBIR project involves integrating the MPMICE code, DYNA3D (for tightly coupled simulations), LS-DYNA (for uncoupled simulations) and a reduced order model (designed for computational efficiency). The following sections describe the MPMICE code in detail along with soil models that have been integrated as part of the SBIR work scope. Details of how MPMICE is being coupled to the DYNA3D and LS-DYNA codes are also discussed.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 NOV 2006	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Software Framework For Blast Event Simulation	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Reaction Engineering International Salt Lake City, UT 84101**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002075., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**8**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

2.1 MPMICE Code

A key model available in the BCF is University of Utah's MPMICE code, which is used primarily to model the blast and soil. MPMICE is a state-of-the-art, multi-material, fluid-structure interaction code, developed as part of the DoE ASCI program at the University of Utah (C-SAFE). Application of MPMICE in the C-SAFE program for modeling explosives and their interaction with solid structures has demonstrated potential for improved accuracy in simulating the explosive-soil-gas interaction within this project. As part of the Phase II project, MPMICE is being enhanced with more advanced constituitive soil models, and it has been time-step coupled to Dyna3D to provide finite-element modeling of the vehicle under study. An MPMICE-based blast preprocessor has also been created to allow uncoupled blast-structure simulations using commercial finite-element packages such as the LS-Dyna software.

MPMICE is an Eulerian compressible CFD code based on a variant of the ICE (Implicit Continuous Eulerian) algorithm first developed by Harlow at LANL [Harlow, 1968]. The variant of ICE being using is a cell-centered, finite volume, multi-material formulation. By multi-material, we mean that an equation for the conservation of mass, momentum and energy is solved for each of the materials involved in the simulation. Included in these governing equations are terms that describe interactions between the various materials. It is through these interaction terms, as well as a multi-material equation of state, that the different materials influence each other.

In order to achieve fluid structure interaction capabilities, advantage is taken of the multi-material nature of the CFD formulation, which allows treatment of the governing equations for any type of material, either fluid or solid. Thus, a simulation of the type targeted in this proposal might include the following materials: air, soil, explosive, explosive products, and steel. In order to avoid the problems that are typically associated with representing solid materials in an Eulerian method (principally diffusion of interfaces between solid and fluid materials) the solid structure will be represented using one of two Lagrangian frame descriptions. The first of these is the Material Point Method, or MPM, [Sulsky, 1995] in which solids materials are represented by a distribution of discrete particles, or material points. State information, (e.g. mass, velocity, temperature, etc), from the solid, as represented by the particles, is projected to the computational grid used in the multi-material CFD formulation, and as such the solid becomes one of the materials whose evolution in time can be computed. Once the solid material's state is advanced in time in the grid representation (simultaneous with the evolution of the fluid material's state), changes in state are interpolated back to the particles to achieve the updated state in the solid. This manner of simulating fluid-structure interaction scenarios has been successfully implemented.

Simulation of the detonation of the HE materials is currently handled within the multi-material ICE formulation through the use of a reactive flow model, namely JWL++ [Souers, 2000]. The JWL++ model describes the rate at which the mass of HE material is converted to a separate material that represents the products of combustion. In addition to JWL++, so-called "program burn" models are under development for use with ideal explosives, such as PBX-9501.

The sequence of images shown in Figure 1 show a steel container filled with PBX9501 (a plastic bonded explosive) that is preheated to the ignition temperature of the explosive. As the explosive reacts, the product gases of the reaction pressurize the container, causing it to bulge, and eventually rupture. As the container ruptures, the products of the reaction escape the container walls through the cracks and eventually engulf the container fragments. This simulation demonstrates several important capabilities. The ability to do reactions involving solid to gas phase change, the ability to treat fragmentation of steel at high strain rates, and, most uniquely, the ability to have gases (the air outside the container, and the products of reaction) that are initially in

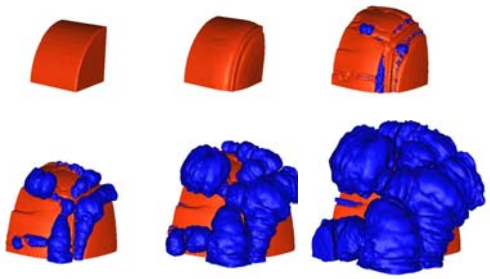

Fig 1. Rupture of steel container

separate regions of the domain, coming into contact.

Both the ICE and MPMICE formulations are implemented within the Uintah Computational Framework (UCF), developed at the University of Utah's Center for the Simulation of Accidental Fires and Explosions (C-SAFE) [www.csafe.utah.edu]. The UCF was specifically designed as a general framework for performing large scale simulations as well as tools for visualizing large datasets. It provides a platform for incorporating a variety of physics algorithms into a computing environment that supports both MPI and thread based parallelism [Rawat, 2001 and Parker, 2002]. In the UCF a wide range of data structures are available to the application programmer, including Particle, Cell, Node and Face centered data on a structured grid.

Domain decomposition is the foundation of the UCF wherein the computational domain is divided into individual "patches" upon which the algorithm is performed. To hide the complexities of parallel data management from the researcher, the UCF incorporates three key features; 1) a task graph representation of the algorithm, 2) a component known as the Data Warehouse, and 3) a Scheduler component. The Data Warehouse acts as a global, single assignment memory abstraction with automatic data lifetime management and storage reclamation. This warehouse automatically handles I/O, check-pointing and restarting. The application developer's algorithm is described by "tasks" where each task corresponds to a single major algorithmic step. Each task contains a list of required input variables and a list of

computed variables. The input requirements have information on data dependencies on neighboring patches. The Scheduler component arranges these tasks into a task graph representation of the algorithm. Data flow among the patches and tasks are represented as "edges" on the graph. From the application developer's perspective, tasks to do this communication are generated automatically, and are scheduled along with those tasks specified by the researcher.

To use the UCF for any particular application, a researcher defines the algorithm on a single patch with tasks. In those tasks the data requirements must be specified, including the number of "ghost" cells needed from the neighboring patches. Once the tasks are specified, the Scheduler creates a task graph, and then distributes the tasks to the available computational resources. Since communication tasks are automatically scheduled and executed, retrieval of data from neighboring patches is transparent to the developer.

The UCF represents the state-of-the-art in high performance scientific computing frameworks, and is Common Component Architecture (CCA) compliant [CCA Forum, CCA Background]. CCA is a component architecture specifically designed for high-end scientific applications. The CCA-compliance of the UCF-based MPMICE software provides significant advantages over legacy codes. Compliance allows the software to rapidly evolve with improvements in the scientific computing community by making use of next-generation scientific components from researchers around the world.

MPMICE is a tool with unique capabilities for modeling container dynamics (such as land mine detonation). In particular, the approach is superior to traditional Arbitrary Lagrange-Eulerian techniques for modeling a situation where two distinct fluid regions are separated by a solid region that fails, allowing the fluid regions to come together into intimate contact. These advantages will provide immediate benefit in the Phase II work where MPMICE is used to model the explosive and soil. This advanced simulation capability provides credence to this SBIR work effort.

Although a viable shell model does not currently exist for MPM, development of such a model is an active research area under the University of Utah's CSAFE program. Project team members will follow the progress of the MPM shell model to determine its applicability to this work. With a viable shell model, it would be possible to simulate an entire vehicle with MPM thus eliminating the need for an FE code. Even without a shell model, MPM is fully capable of modeling armor plate and land mine casings where wall thicknesses are large enough to allow computationally tractable meshing. Given the uniqueness of the MPM approach to the proposed work, it is important to define some of its advantages when compared to FE approaches:

- No mesh distortion or mesh entanglement issues. This is particularly helpful in simulations involving a lot of deformation. FE meshes can become inverted, tangled, or squashed down to a poor aspect ratio, any of which will cause difficulties with the simulation.
- Creation of initial geometry. Because a body fitted mesh isn't used, generating a representation of the geometry using particles is straight forward, and can be accomplished using a number of different methods. With any of these methods, generating the particle description takes minutes vs. hours or days to generate an FE mesh.
- Contact between bodies. Because MPM uses a structured grid, upon which we maintain separate velocity data for as many distinct objects as needed, contact interactions can be done on a node by node basis with no requirement to maintain information about surfaces. This is one of the more complex problems that must be addressed in FE codes.

2.2 Soil Modeling

To address the need for soil modeling capabilities within the MPMICE code, three different soil models have been implemented to date. The first is the Tillotson equation-of-state [Hill, 1970], [Tillotson, 1962], the second is the Gruneisen equation-of-state [Woodruff, 1973] and the third is the Soil and Foam constituitive model known as Material 5 in the LS-DYNA software [Krieg, 1972]. In MPMICE, constitutive models providing a description of the deviatoric stress behavior of materials (such as the Soil and Foam model) must be modeled in the MPM side of the code. Since MPM has greater computational requirements than ICE, computational savings may be obtained by modeling the soil with an EOS such as Tillotson or Gruneisen. Each of the three models is described in the following sections.

Tillotson: After reviewing a number of equation-of-state-based models, the Tillotson model was chosen as an initial proof-of-concept model for the Phase I effort. This model has been used or considered in previous studies [Lottero and Kimsey, 1978], [Bessette, 2003] and its limitations and capabilities are well documented [Gathers, 1994]. The Tillotson equation-of-state (EOS) is expressed as a two-regime expression. In the compressed state the pressure is expressed as Equation 1:

$$P = \left[a + \frac{b}{\frac{I}{I_0 \eta^2} + 1} \right] I\rho + A\mu + B\mu^2, \ \eta \geq 1 \qquad (1)$$

where η is the ratio of the soil density to reference density, ρ/ρ_0, $\mu = 1- \eta$, I is the internal energy, and I_0 is the reference internal energy. The expression for expanded states is given by Equation 2:

$$P = aI\rho + \left[\frac{bI\rho}{\frac{I}{I_0 \eta^2} + 1} + A\mu e^{-\alpha\left(\frac{\rho_0}{\rho} - 1\right)} \right] e^{-\beta\left(\frac{\rho_0}{\rho} - 1\right)^2}, \ \eta < 1 \qquad (2)$$

a, b, A, B, α,and β are constants. The values used in this model are taken as

a = 0.5
b = 1.3
A = 0.0 GPa
B = 3.0 GPa
I_0 = 6.0 MJ/kg
α =5.0
β =5.0

The constant A was taken as zero since over a range of densities lower than the reference density the pressure becomes negative, which is non-physical and cannot be handled in the MPMICE algorithm. Figure 2 shows the pressure as a function of density from the above EOS. In Figure 2a the pressure is plotted with the constant A taking the value recommended by Lottero and Kimsey [1978]. The negative pressure is seen at densities between about 1000 kg/m^3 and the reference density of 1700 kg/m^3. Figure 2b shows the relationship with A = 0. The internal energy was chosen such that at atmospheric pressure the reference density was recovered. In these simulations the internal energy was treated as constant.

The Tillotson EOS was added to the EquationOfStateFactory within the ICE portion of MPMICE. The EquationOfStateFactory class is the avenue for adding equations of state to the MPMICE software.

Since the soil is modeled as an EOS no deviatoric stresses are modeled. Therefore, the MPM formulation was not required to represent the soil within MPMICE. More sophisticated soil constitutive models require MPM representation.

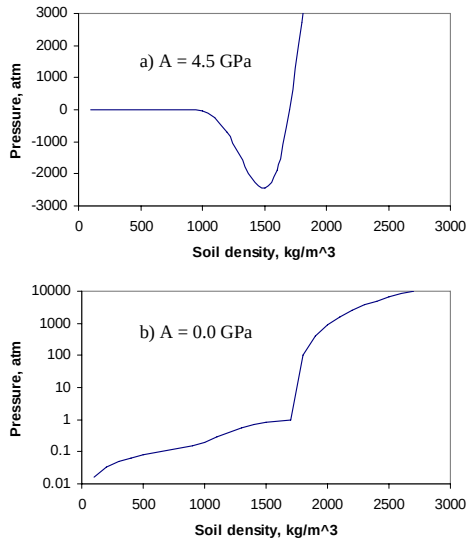

Fig. 2. Tillotson EOS pressure vs. soil density.

Gruneisen: The Tillotson model was demonstrated during the Phase I effort. However, that model required some manipulation so that unphysical negative pressures were avoided. As an alternative to Tillotson, the Gruneisen EOS model [Woodruff, 1973] was implemented. The pressure p of this model is expressed for compressed materials ($\mu \geq 0$) as Equation 3:

$$p = \frac{\rho_0 C^2 \mu \left(1 + \left(1 - \frac{\gamma_0}{2}\right)\mu - \frac{a}{2}\mu^2\right)}{\left[1 - (S_1 - 1)\mu - S_2 \frac{\mu^2}{1+\mu} - S_3 \frac{\mu^3}{(1+\mu)^2}\right]^2} + (\gamma_0 + a\mu)E \qquad (3)$$

and for expanded materials ($\mu < 0$) as Equation 4:

$$p = \rho_0 C^2 \mu + (\gamma_0 + a\mu)E \qquad (4)$$

where E is the internal energy per unit volume, C is the intercept of the shock velocity vs. particle velocity curve, S_1, S_2, and S_3 are the coefficients of the slope of that curve, γ_0 is the Gruneisen gamma, and a is the first order volume correction to γ_0. Each of these coefficients are input parameters to the model. The excess compression is defined by Equation 5:

$$\mu = \frac{\rho}{\rho_0} - 1 \qquad (5)$$

where ρ is the current density and ρ_0 is the initial density.

If we take S_1 =1 and S_2 = S_2 = 0, the coefficients of this model may be obtained from a cubic polynomial fit of pressure versus volumetric strain data from a hydrostatic compression test.

Soil and Foam: The Soil and Foam constituitive model has been implemented into the MPM portion of MPMICE. This constitutive model allows for three basic modes of soil deformation and failure. The first is a large amount of compaction due a relatively high, in situ, void fraction. The second is the ability to allow gas venting for an explosive insufficiently buried for strain energy to contain the energy of the blast. In this mode the venting of the explosive gases occurs when the soil fails, and an open path develops between the surface and the contained explosive gases. Finally, since soils have very low tensile strengths, the soil near the surface will be ejected, at relatively high velocity.

If sufficient material characterization data is available, the model parameters can be calibrated to the data. If insufficient characterization data is available, the model parameters will need to be estimated and the fewer parameters the more likely a consistent set of estimates can be achieved.

In the Soil and Foam model a pressure-dependent flow rule governs the deviatoric behavior. This shear failure surface is given as a quadratic of the pressure p as follows in Equation 6:

$$J_2' = a_0 + a_1 p + a_2 p^2 \qquad (6)$$

where J_2' is the second deviatoric stress invariant at yield.

Volumetric yielding is determined by a tabulated curve of pressure versus volumetric strain. The maximum volumetric strain in compression is stored. If the new compressive volumetric strain exceeds the stored value, loading is indicated. When the yield condition is violated, the updated trial stresses are scaled back.

The input parameters required by the model are:

- The elastic shear modulus
- The bulk unloading modulus
- The three coefficients a_0, a_1, a_2 defining the deviatoric yield as a function of pressure
- The curve points defining the pressure versus volumetric strain

2.3 Coupling to Finite Element Models

As part of the SBIR work scope, MPMICE is being coupled to the DYNA3D and LS-DYNA models. This approach allows MPMICE to model the blast and soil, while the DYNA codes handle the finite-element response of the vehicle under study. Although MPMICE is capable of modeling a complete blast+structure scenario (with the structure modeled using MPM), the computational requirements are currently intractable for complex, thin walled geometries such as vehicles. Hence is the need for utilizing the FE codes to handle this portion of the calculation. In the following sections we describe the approaches to coupling the codes in detail.

Tightly Coupled MPMICE-to-DYNA3D

The major focus in the Phase I effort was to define and implement a coupling between MPMICE and DYNA3D. The detailed plan developed to accomplish the coupling included what physical quantities are exchanged, when in the algorithm the data exchange occurs, how time-stepping is controlled, and how material data consistency is maintained between the two codes.

The physical quantities that must be exchanged between MPMICE and DYNA3D are highlighted in Figure 3. The data MPMICE requires from DYNA3D include mass, momentum and internal energy. The data that DYNA3D requires from MPMICE are the forces (stresses) exerted on the structures. The data exchange is accomplished using the MPM particles contained used within MPMICE. These particles coincide with the nodes in the FE model and thus make for a natural and convenient mechanism to exchange the required data. In addition, using MPM particles allows use of the existing interpolation particle-cell-center machinery that already exists in MPMICE. Initially it was proposed that the data exchange would occur directly through the Eulerian mesh of ICE. However, after further analysis of the problem, the team decided that the best solution was to create MPM particles that are coincident with the DYNA3D finite element nodes. This greatly simplifies data transfers because the particles are free to move with the mesh.

By using the coincident mapping of MPM particles to FE nodes, the algorithm for performing a time-step can be described as follows (see also Figure 3):

1. Project the vehicle material state data from the unstructured DYNA3D mesh to the corresponding MPM particles in the MPMICE code. State data includes mass density, momentum and internal energy.

2. Time advance the multi-material equations of motion in MPMICE, where the material representation of the vehicle interacts with the other materials through exchange terms in the momentum equation. In addition, the gradient in the pressure field also provides a source of momentum to the vehicle materials.

3. Convert the momentum sources of the vehicle material from step 2 into accelerations and interpolate these data back to the MPM particles.

4. Transfer the data from the MPM particles to the unstructured DYNA3D mesh.

5. Time advance the solution of the vehicle using DYNA3D to update its state and geometric representation.

6. Complete the time-step in the MPMICE code, advancing the solution of those materials represented in the Eulerian frame as well as the solid materials represented using MPM.

7. Continue to next time-step.

As part of the Phase II efforts, this coupling strategy is being modified based on the findings from Phase I. Specifically, the treatment of complex geometries associated with vehicles is required. Definition and construction of MPM particles for complex geometries within the MPMICE code are accomplished via triangulated surface inputs. Those surfaces can be generated from the FE representation of the vehicle model. This approach has necessitated that the MPM particle locations not coincide with the FE nodes as was required in the software demonstrated in the Phase I effort. Also in the Phase II effort loads at load segments are passed to DYNA3D from the MPMICE solution rather than accelerations at nodes. This is anticipated to reduce the hour glass effect in the DYNA3D solution.

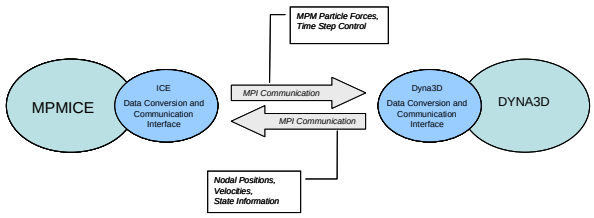

Fig. 3. MPMICE, DYNA3D Software Coupling

Uncoupled MPMICE-to-LS-DYNA

In addition to the time-step-level coupling implemented between MPMICE and DYNA3D, the team has also implemented a coupling between MPMICE and LS-DYNA (no time-step data exchange). A post-processing coupling was chosen in this instance since the team has no source code access to LS-DYNA. In this

approach the vehicle geometry remains rigid in the MPMICE simulation followed by a load curve extraction from the MPMICE solution to be used in a separate LS-DYNA simulation.

2.4 Reduced Order Modeling

Performing detailed modeling of the blast event and the surrounding soil can require substantial computational resources. This class of simulations are typically targeted for supercomputer-level computer hardware and the computational models require expert level personnel to setup, run and post-process the simulation.

However, there is a broad spectrum of problems to address in blast simulation. More importantly, not all blast simulations require the high-fidelity and complex physics typically included in these high-fidelity simulations. There is a significant need for computational tools that can be used on standard personal computer level hardware. Such tools would provide the ability for a non-expert user to perform scoping-level studies, with reasonable accuracy in a short time period.

As part of the SBIR project, REI is in the process of evaluating a number of different approaches for creating a reduced order blast+soil model. These include the method of Proper Orthogonal Decomposition (POD), various statistical modeling methods, and full-physics approaches based on MPMICE that involve reduced mesh sizes and adaptive mesh refinement. Computationally efficient methods found to be useful will be integrated with the BCF.

3. COMPUTATIONAL FRAMEWORK

At present, detailed modeling of the blast-vehicle-crew scenario requires the use of a number of disparate simulation codes. Coupling these codes together typically involves writing special purpose data converters. In addition, if the simulation tools are restricted to specific computer platforms, then all data handling between the codes must be done manually or using ad-hoc techniques.

To resolve these difficulties, REI is developing a Blast Computational Framework. The goal of the BCF will be to provide a unified simulation environment where the modeling tools required for the blast-to-crew simulation can be integrated and interoperate with one another. A user can then graphically construct a simulation by choosing from a library of available models. The framework handles the details of generating inputs for the models, moving information between the models and controlling execution scheduling. In addition, the framework allows the models to be run on disparate computing platforms and provides post-processing and visualization capabilities.

As part of the scope of work for the project, the BCF will provide tools for performing soil bound explosion simulations, their effects on selected vehicle geometries and on the human occupants of the vehicles. The extensible architecture provided by the BCF make it possible to integrate additional models in future efforts.

BCF Software Architecture

Figure 4 shows the software architecture of the BCF. To integrate models with the BCF, plug-in components are created, which encapsulate computational models and other tools that are to be accessed from the framework. Each component has user interface input panels for configuration and control functions. The components are integrated into the framework using a standardized CORBA [www.omg.com] component architecture interface definition language specification. It should be noted that the software infrastructure within the framework imposes no limitations on the quantity or type of data that can be passed between component models, the programming language used to create a model or the physics and numerical methods used by a model. The framework supports heterogeneous, distributed computing (via CORBA functionality) and thus allows models to be executed on remote computing resources (e.g., single computers or clusters accessible via a network).

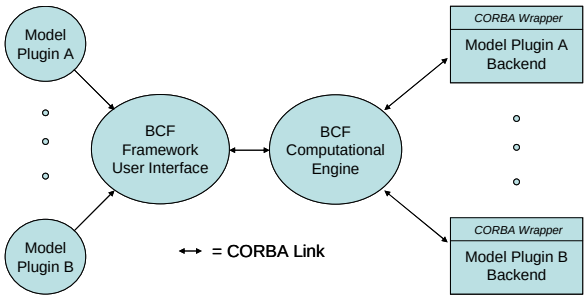

Fig. 4. BCF Software Architecture

4. DEMONSTRATION CALCULATIONS

In this section, we provide examples of simulations performed during the course of the SBIR project. The focus thus far has been on relatively simple geometries where experimental data is available for validation.

4.1 Tightly Coupled MPMICE-DYNA3D

This test case was used in Phase I to demonstrate tightly coupled MPMICE-DYNA3D. The configuration is similar to that of Gupta [Gupta, 1999] and involves a 10 kg charge, 40 cm stand-off and 15 cm depth-of-burial. The Tillotson EOS was used for soil modeling. The 20cm thick plate was modeled in DYNA3D with hex elements. The ICE mesh for the simulation was 100x350 and the total simulation time was ~36 hours on two AMD Operton processors. Figures 5 shows the simulation results at a time of approximately 1.5 ms. This figure shows the soil colored by density and the plate colored by velocity. Comparison of the plate tilt angle as a function of time between the simulation results and those of Gupta showed reasonable agreement.

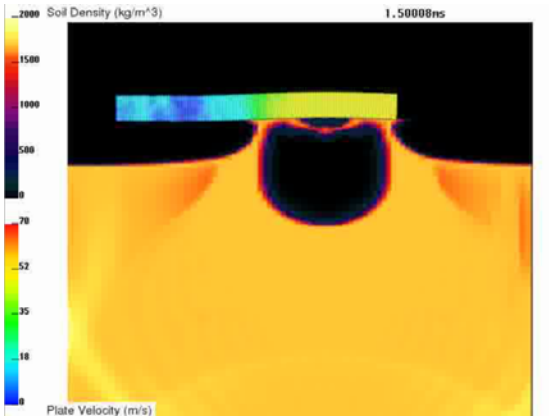

Fig. 5. Simulation with tightly coupled MPMICE-DYNA3D.

4.2 Standalone MPMICE

In this case MPMICE was used to model the thin plate geometry reported by Williams and McClennan [2002] in a Defence R&D Canada (DRDC) paper. In that study they compared modeled predictions of plate displacement with measurements. A 2D version of this configuration was modeled to demonstrate the tightly MPMICE-Dyna coupling of the Phase I effort [Swensen, 2005]. Two symmetry planes along the plate centerlines were used. The 3-dimensional grid consisted of 460,000 cells. Soil modeling was performed using the Soil and Foam MPM constitutive model. The case took 5 days on a cluster of six 64 bit computers. Figure 6 shows the results at the time of approximately 1.5 ms. The isosurfaces in the figure were generated from soil density and then colored by velocity. Figure 9 presents the displacement of a point on the plate 6 inches from the center compared to measurements reported by Williams and McClennan [2002]. Also shown in Figure 9 is the prediction using LS-DYNA where the blast and soil are modeling using SPH. As seen Figure 9, there is good agreement between MPMICE and the measured data.

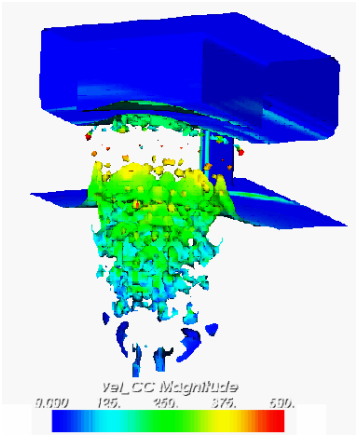

Fig. 6. Simulation of DRDC Case using full MPMICE.

4.3 Decoupled MPMICE-LS-DYNA

This case again involves a simulation of the DRDC plate configuration. The simulation was performed using the BCF with MPMICE acting as a blast+soil preprocessor and LS-DYNA modeling the response of the plate. Figure 7 shows the BCF network diagram used for the simulation. Figure 8 shows the plate colored by pressure at ~2.3ms. Figure 9 again shows the displacement of a point on the plate 6 inches from the center compared to measurements reported by Williams and McClennan [2002]. As seen in this figure, the decoupled simulation over-predicts the plate displacement. This shows the importance of tight data coupling for the geometry under study.

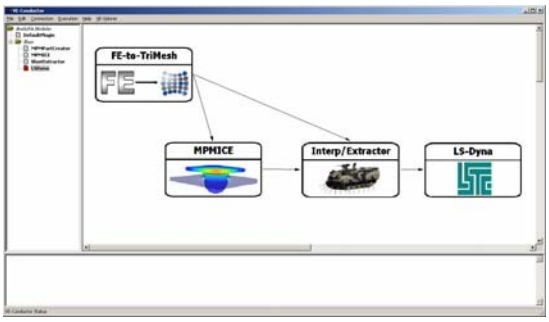

Fig. 7. BCF dataflow network for a decoupled MPMICE, LS-DYNA simulation of DRDC.

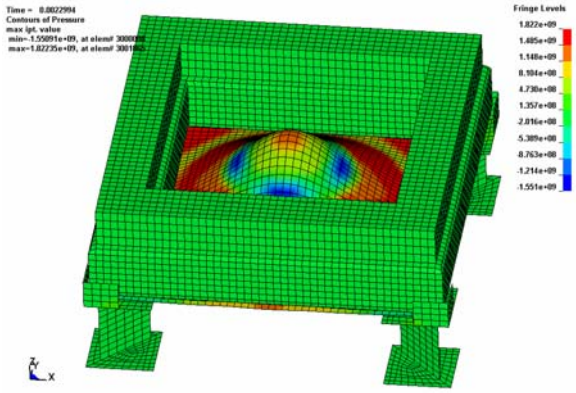

Fig. 8. Simulation results for BCF decoupled MPMICE, LS-DYNA simulation of DRDC.

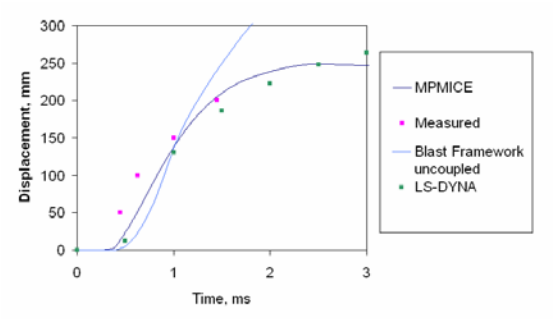

Fig. 9. Comparison of DRDC plate displacement for various models.

CONCLUSION

This paper has described an ongoing effort, funded through the Army SBIR program, to develop a state-of-the-art computational framework for blast-vehicle-crew simulations. The framework will provide an advanced, unified simulation environment where the various disparate computational models required for a complete blast-to-crew analysis can seamlessly interact with one another, eliminating the complex and inefficient manual coupling methods now used for multi-model simulations. The framework is being designed to make use of the latest technologies in component-based software design, which will enhance functionality and extensibility.

In addition to the core framework development, the project is also working to extend the state-of-the-art DoE ASCI funded MPMICE code to the class of problems under study. This code offers a number of advantages when compared to traditional blast-structure simulation techniques and offers the possibility of attaining the next level of accuracy in these types of simulations.

The objective of the project is for the enhanced simulation capabilities provided by the BCF and component models to better protect the crews of existing vehicles and to help design next-generation vehicles as well.

ACKNOWLEDGEMENT

Funding for this project is provided through a DoD-Army SBIR Phase II, “A Software Framework For Blast Event Simulation”, contract number: W56HZV-05-C-0630; Program Official: Mr. Richard Goetz (RDECOM-TARDEC). The authors would like to acknowledge the contributions of Dr. Steven Kirkpatrick and Brian Peterson (Applied Research Associates) for their assistance with LS-DYNA. We would also like to acknowledge the support of Dr. Len Schwer (Schwer Engineering and Consulting) for his assistance with soil modeling issues.

REFERENCES

Bessette, G. C., Vaughan, C.T., Bell, R.L. and Stephen W. Attaway, S.W., 2003, “Modeling Air Blast on Thin-shell Structures with Zapotec” www.cs.sandia.gov/PublicationsPresentations/2003/Bessette_2003_3420.pdf.

CCA Background, http://www.cca-forum.org/tutorials/archives/2002/tutorial-2002-04-10/ccatut-cca-and-ccttss.pdf.

CCA Forum, http://www.cca-forum.org.

Gathers, R., Selected Topics in Shock Wave Physics and Equation of State Modeling, World Scientific, 1994.

Global Security, http://www.globalsecurity.org/military/systems/ground/iav.htm.

Gupta, A., "Estimation Of Vehicle Floor Plate Loading And Response Due To Detonation of a Mine Shallow-Buried in Dry Sand and Wet Tuff," US Army Ground Vehicle Survivability Symposium, Monterey, CA, 29 Mar-1 Apr 1999.

Harlow, F. H.; Amsden, A.A. Numerical Calculation of Almost Incompressible Flow. J. Comp. Phys. 1968, 3, 80-93.

Hill, L.R. and Johnson, W.E., Energy Partitioning during Hypervelocity Impact on Rocks,SC-R-70-4402, Sandia Laboratories, Albuquerque, NM (December 1970).

Krieg, R.D., “A Simple Constituitive Description for Cellular Concrete,” Sandia National Laboratories, Albuquerque, NM, Rept. SC-DR-72-0883 (1972).

Lottero, R.E., and Kimsey, K.D., 1978, A Comparison of Computed Versus Experimental Loading and Response of a Flat Plate Subjected to Mine Blast, USA Ballistic Research Laboratories Report ARBRL-MR-02807.

Parker, S. “A Component-Based Architecture for Parallel Multi-Physics {PDE} Simulation.” Presented at the International Conference on Computational Science (ICCS2002), April 21-24, 2002.

Rawat, R.; Parker, S.G.; Smith, P.J.; Johnson, C.R. “Parallelization and Integration of Fire Simulations in the {Uintah PSE}.” Presented at the Tenth SIAM Conference on Parallel Processing for Scientific Computing, 2001.

Souers, P. C.; Anderson, S.; Mercer, J.; McGuire, E.; Vitello, P. JWL++: A Simple Reactive Flow Code Package for Detonation. Propellants, Explosives and Pyrotechnics 2000, 25, 54-58.

Sulsky, D.; Chen, Z.; Schreyer, H. L. A Particle Method for History-Dependent Materials. Comp. Methods Appl. Mech. Engrg. 1994, 118, 179-196.

Swensen, D., “Development of Blast Event Simulation” SBIR Phase I Final Report SBIR Contract No. W56HZV-05-C0115, 2005.

Tillotson, J.H., Metallic Equations of State for Hypervelocity Impact,GA-3216, General Atomic, Division of General Dynamics, San Diego, CA (July 1962).

Williams, K. and McClennan, S., "A Numerical Analysis of Mine Blast Effects on Simplified Target Geometries: Validation of Loading Models," Defence R&D Canada, DRDC Valcartier TM 2002-260, 2003.

Woodruff, J.P., "KOVEC User's Manual," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-51079, 1973.